

Report on

# **RideHive - Bike and Scooter Rental System**

- By

Samruddhi Chavan | N01604191

Saloni Patel | N01603895

Sruthi Pandiath | N01618202

Mitali Sisodia | N01621572

## Table of Contents

1. Introduction .....	3
1.1. Project Overview .....	3
1.2. Objectives .....	3
2. Problem Statement .....	3
3. Proposed Solution .....	3
4. Technologies Used .....	4
5. Database Schema .....	4
6. Features .....	5
7. UML Diagrams .....	6
7.1. Class Diagram .....	6
7.2. Sequence Diagram .....	8
8. Deployment Process Using Podman .....	10
9. Screenshots .....	11
10. Conclusion .....	31

# 1. Introduction

RideHive is a web-based bike and scooter rental system designed to provide affordable and flexible transportation for urban users. It streamlines vehicle rentals for users, vendors, and admins with features like secure payments, efficient fleet management, and modular scalability.

## 1.1. Project Overview

**RideHive** is a web-based platform developed to provide convenient access to short-term bike and scooter rentals. Catering to users, vendors, and administrators, the system leverages Laravel and other technologies to ensure functionality, security, and scalability. By addressing urban transportation needs, RideHive offers a streamlined solution for affordable and flexible vehicle rentals.

## 1.2. Objectives

The primary objectives of the RideHive platform are as follows:

- Simplify the process of renting bikes and scooters for users.
- Empower vendors to efficiently manage vehicle listings and bookings.
- Facilitate administrators in monitoring and regulating platform operations.
- Ensure secure and user-friendly online payments through **Stripe integration**.
- Develop a modular and scalable architecture to handle future enhancements.

# 2. Problem Statement

With the increasing demand for temporary and cost-effective transportation solutions, RideHive fills a critical gap in the market. Urban dwellers, students, tourists, and professionals benefit from a flexible rental service that does not require vehicle ownership.

# 3. Proposed Solution

- **Urban Need:** In cities, many individuals require transportation for short durations. Renting is often more practical and economical than ownership.
- **Vendor Opportunity:** Vendors can list underutilized vehicles, generating passive income.
- **Streamlined Operations:** By centralizing transactions, bookings, and listings, RideHive ensures ease of use for all stakeholders.
- **Technology-Driven Efficiency:** The platform's Laravel-based backend and Stripe integration provide a secure, responsive, and scalable solution.

## 4. Technologies Used

- ❖ PHP: Core programming language for backend logic.
- ❖ MySQL: Database management system.
- ❖ HTML/CSS(Tailwind)/JavaScript: Frontend technologies for user interface.
- ❖ Autoloading: Composer facilitates class autoloading, minimizing boilerplate code and ensuring efficient application setup.
- ❖ Database Migration: Laravel's migration system manages database version control, making schema changes collaborative and conflict-free
- ❖ Laravel is a PHP-based framework that adheres to the Model-View-Controller (MVC) architecture, ensuring clean code and modular development.

## 5. Database Schema

- ❖ **Users Table**
  - Fields: user\_id, name, email, password, role, status, created\_at
  - Relationships: One-to-many with bookings and feedback.
- ❖ **Vehicles Table**
  - Fields: vehicle\_id, vendor\_id, type, model, price\_per\_day, status, created\_at
  - Relationships: Belongs to vendors; one-to-many with bookings and availability.
- ❖ **Bookings Table**
  - Fields: booking\_id, user\_id, vehicle\_id, start\_date, end\_date, total\_cost, status
  - Relationships: Belongs to users and vehicles.
- ❖ **Payments Table**
  - Fields: payment\_id, booking\_id, amount, status, payment\_date
  - Relationships: Belongs to bookings; ensures secure transactions.
- ❖ **Availability Table**
  - Fields: availability\_id, vehicle\_id, start\_date, end\_date, status
  - Relationships: Tracks vehicle availability, linking to bookings when applicable.
- ❖ **Relationships Summary**
  - Users → Bookings → Vehicles → Payments: Tracks user interaction from browsing to payment.
  - Vendors → Vehicles → Availability: Links vendors with their fleet and rental status.

## 6. Features

RideHive consists of distinct modules designed for different user roles:

### **Admin Module**

- Manage users, vendors, vehicles, and bookings.
- Monitor transactions via dashboards.

### **User Module**

- Search and filter available vehicles by type, price, and availability.
- Book and pay for rentals using Stripe for secure online payments.

### **Vendor Module**

- List, update, and remove vehicles with pricing and availability details.
- Track bookings and customer interactions.

### **Payment Module**

- Integrated with Stripe API, enabling secure and automated payment processing.
- Handles payment status (successful, failed, pending) and logs transaction details.

## 7. UML Diagrams

Unified Modeling Language (UML) is a standardized modeling language used in software engineering. It helps visualize, specify, and document the architecture and design of software systems. UML includes several diagram types such as class diagrams, sequence diagrams, use-case diagrams, and more.

### 7.1. Class Diagram

#### 1. Purpose:

- Represents the structure of the system.
- Shows classes, their attributes, methods, and relationships.

#### 2. Elements:

- **Classes:** Represent entities with attributes (data) and methods (functions).
- **Relationships:**
  - **Association:** Represents "uses" or "has-a" relationships (e.g., Users "owns" Vehicles).
  - **Multiplicity:** Specifies how many instances participate in the relationship (e.g., 0..\* means "zero or more").
  - **Inheritance:** Shows generalization-specialization relationships (not explicitly shown in this diagram).
  - **Aggregation/Composition:** Denotes "part-of" relationships (not shown here).

#### 3. Analysis of the Class Diagram:

- The system manages users, vehicles, bookings, availability, payments, and feedback.
- Key Entities:
  - **Users:** Handles user details and operations like login, registration, and booking.
  - **Vehicles:** Represents rentable vehicles with attributes like type, model, price, and status.
  - **Bookings:** Links users to vehicles through reservation details.
  - **Availability:** Tracks when vehicles are available.
  - **Payments:** Records payment details for bookings.
  - **Feedback:** Stores user reviews for bookings.
- Relationships:
  - Users can own Vehicles.
  - Vehicles have Availability and are linked to Bookings.
  - Bookings are associated with Payments and Feedback.

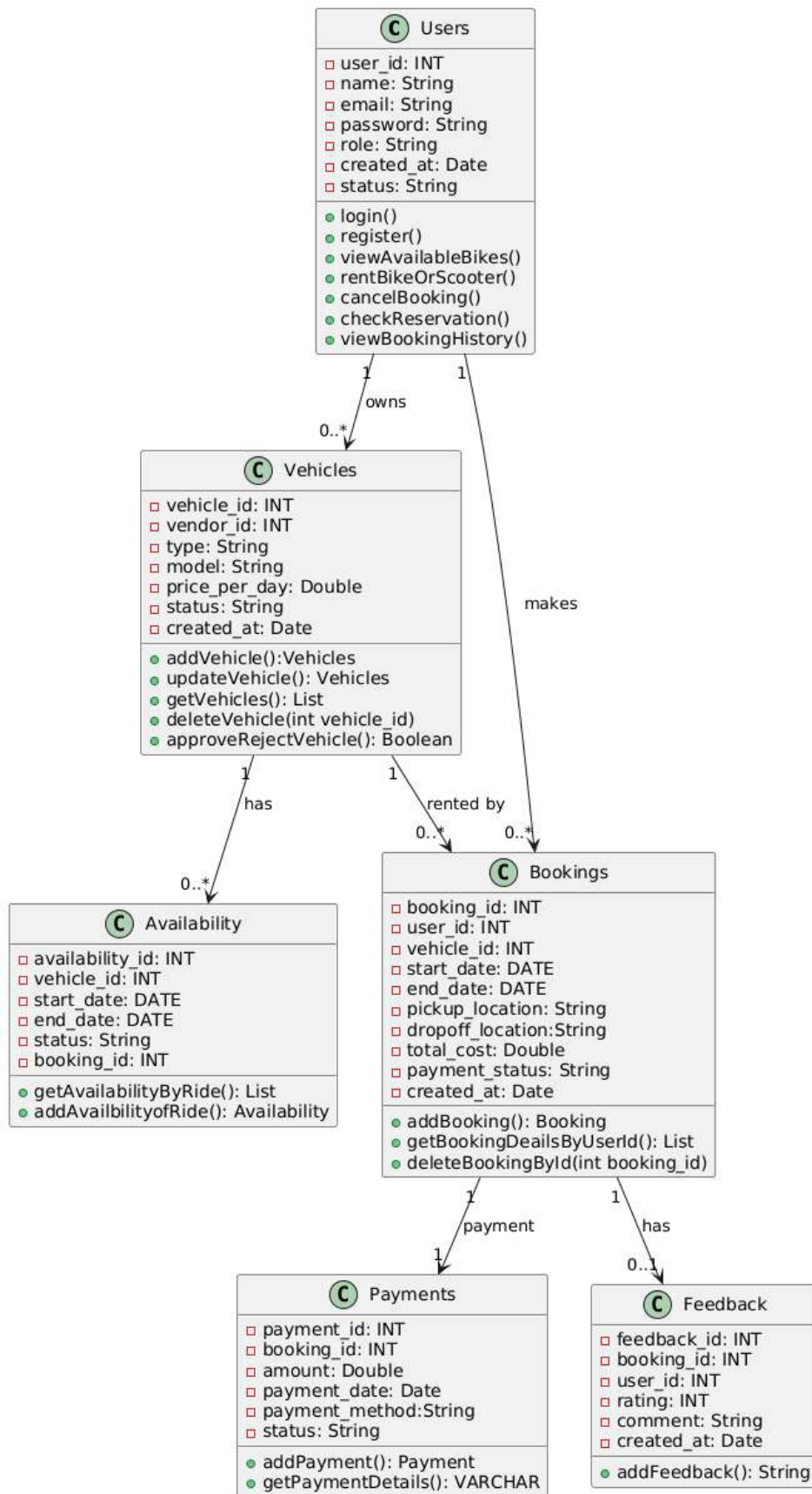


Fig.1. Class Diagram

## 7.2. Sequence Diagram

### 1. Purpose:

- Illustrates how processes or operations are carried out.
- Displays interactions between system components in a time-sequential manner.

### 2. Elements:

- **Actors:** Represent users or systems interacting with the software (e.g., Vendor, User, Admin).
- **Lifelines:** Show the lifespan of an object during interactions.
- **Messages:** Depict communication between objects (synchronous or asynchronous).
- **Execution Occurrence:** Indicates when an object is performing an operation.

### 3. Analysis of the Sequence Diagram (Image 2):

- Key Use Cases:
  1. **User and Vendor Registration and Login:**
    - Users and vendors register and validate credentials via authorization emails.
    - Login is confirmed once credentials are validated.
  2. **Vehicle Management:**
    - Vendors can add, update, and delete vehicle records, which are reviewed for approval.
  3. **Vehicle Booking:**
    - Users search for available vehicles, request bookings, and make payments.
    - Payment status is updated, and booking confirmation is sent.
  4. **Vehicle Availability and Cancellation:**
    - Availability is queried when a user checks for a ride.
    - Bookings can be canceled, updating the vehicle's availability.
  5. **Admin Management:**
    - Admin retrieves payment transactions and user/vendor details for oversight.



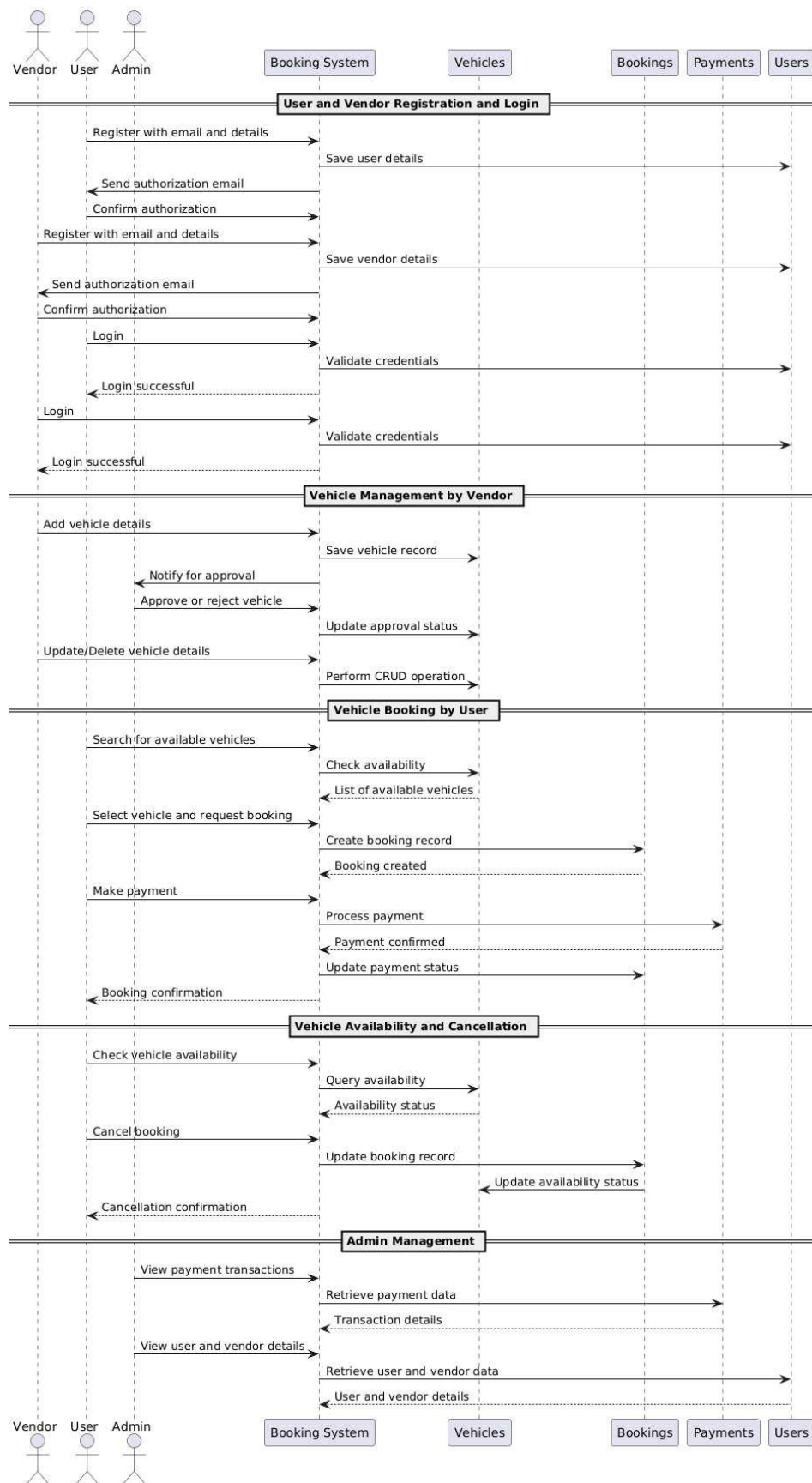


Fig.2. Sequence Diagram

## 8. Deployment Process Using Podman

For deploying our Federal Parliament Legislation Process Management System, we used **Podman** to manage containerized applications due to its enhanced security and compatibility with Docker commands. Podman enables rootless containers, allowing applications to run without root privileges, which improves security.

**Dockerfile:** Defines the application environment, including the PHP version and necessary dependencies.

*FROM php:8.1-apache* – This line creates an image of apache 8.1 to run the php application

*RUN docker-php-ext-install pdo pdo\_mysql* – This line asks the container to install mysql-pdo

*COPY src/ /var/www/html/* - This line copies all the content from src/ to the root directory of container

*COPY .env /var/www/html/.env* - This line copies all the content from .env file to the root directory of container

*WORKDIR /var/www/html* – This line declares this as its working directory

*EXPOSE 80* – This line specify that container will listen on port **80**

### **docker-compose.yml:**

This file defines two main services:

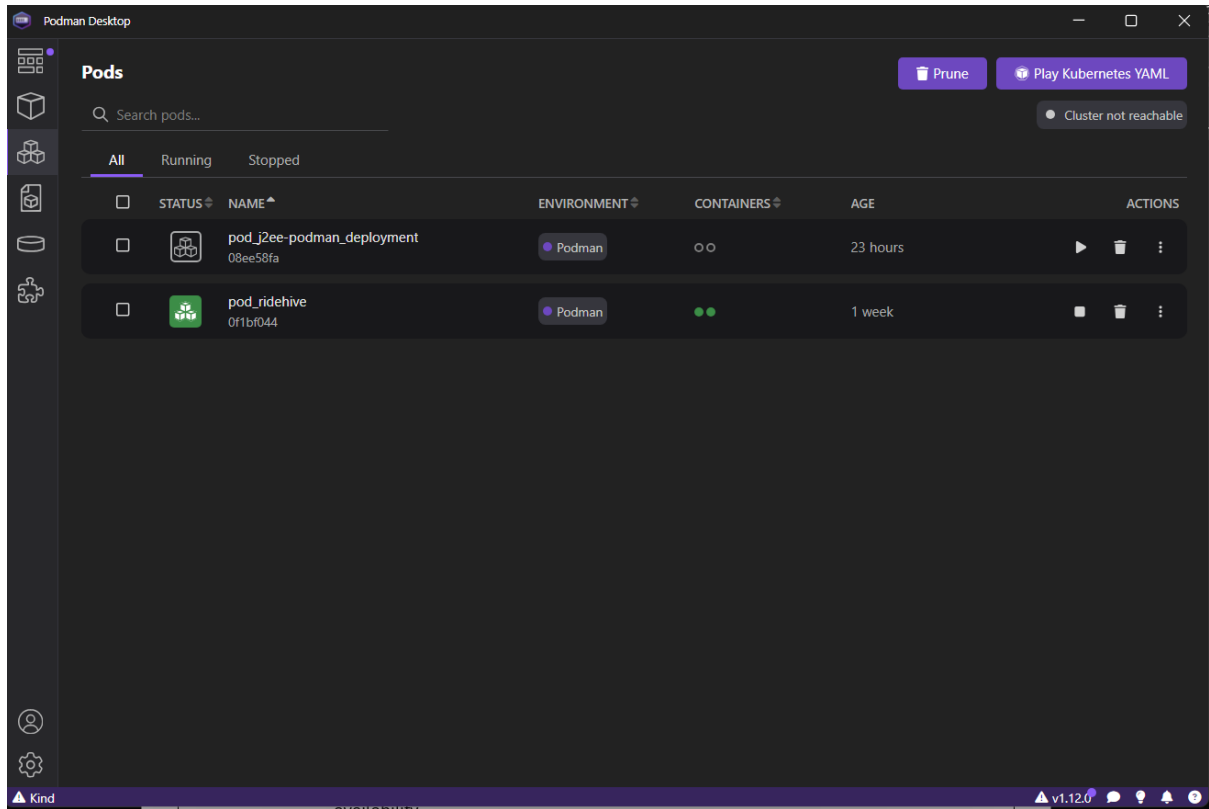
**PHP-Apache:** Handles the PHP application running on an Apache web server, with the project files and environment variables mounted as volumes.

**MySQL Database:** Configures a MySQL 8.0 database, including initialization with SQL scripts and database persistence using volumes.

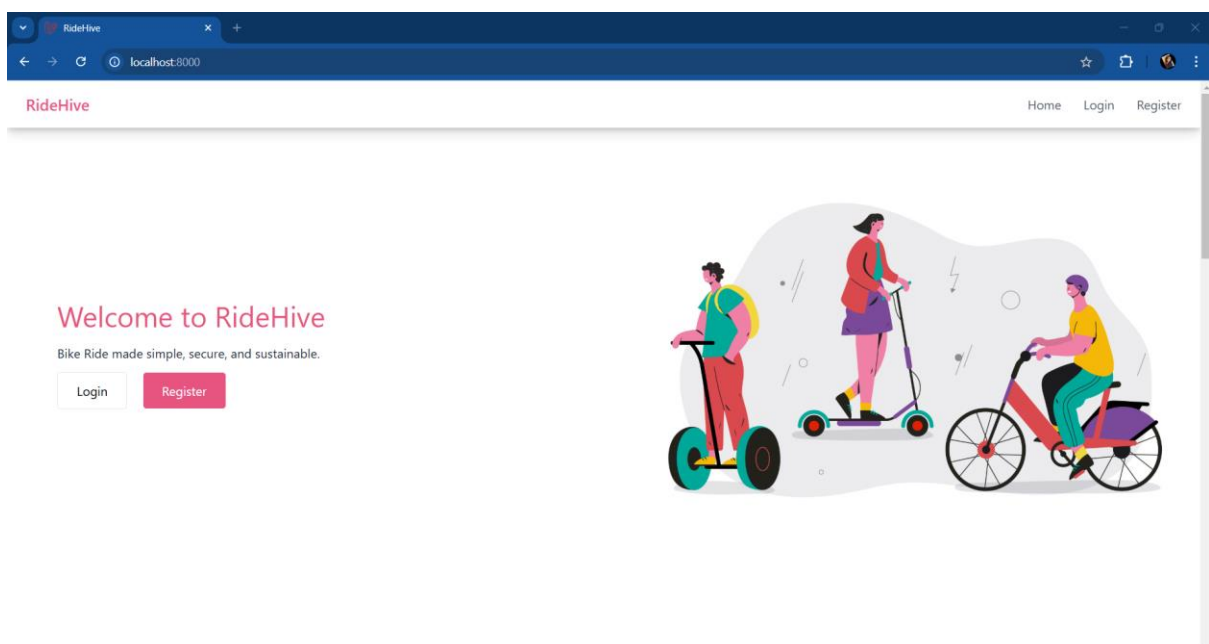
**Dockerfile:** Builds the PHP environment using version 8.1, installing necessary extensions such as pdo and pdo\_mysql for database interaction. The source files are copied into the web server directory, and port 80 is exposed for web access.

## 9. Screenshots

### Podman Deployment:



### Home Page:





## Our Features



### Easy Rental

Rent a bike or scooter effortlessly.



### Save Money

Reduce your transportation costs.



### Eco-Friendly

Help reduce your carbon footprint by riding bikes.



## What Our Users Say



"RideHive made my daily commute so much easier! It's quick, affordable, and eco-friendly."

**Jhon J.**

User for 3+ years



"I love that I can find rental bikes nearby. It's both cost-effective and convenient!"

**Sarah D.**

User for 1+ year



"Being part of RideHive has helped me reduce my travel expenses while helping the environment!"

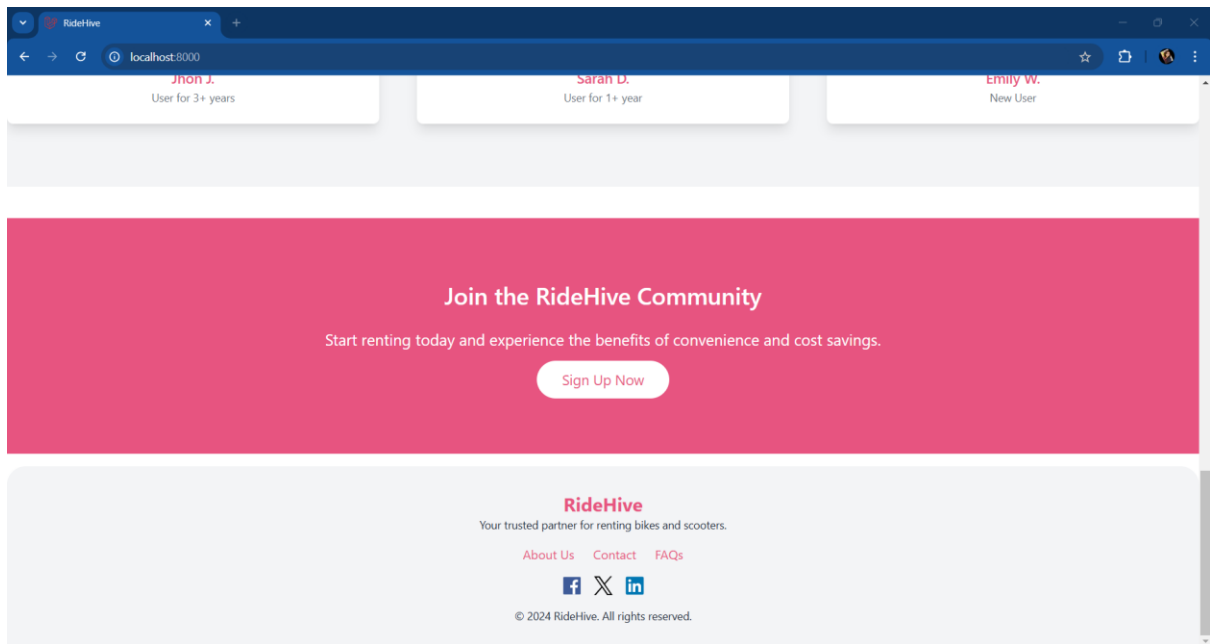
**Emily W.**

New User

## Join the RideHive Community

Start renting today and experience the benefits of convenience and cost savings.

[Sign Up Now](#)



## Register as Vendor:

A screenshot of the RideHive registration form, titled 'Register', located at 'localhost:8000/register'. The form contains the following fields: 'Full Name' with the value 'Jhon Markson', 'Email' with the value 'jhon@gmail.com', 'Contact Number' with the value '4537778595', 'Register As' with a dropdown menu set to 'Vendor', 'Password' with masked characters '\*\*\*\*\*', and 'Confirm Password' with masked characters '\*\*\*\*\*'. A pink 'Register' button is at the bottom of the form. The browser's address bar shows 'localhost:8000/register' and the RideHive logo is visible in the top left corner.

RideHive

Home Login Register

### Login

Email

Password

Login

Don't have an account? Register Now!

**RideHive**  
Your trusted partner for renting bikes and scooters.

[About Us](#) [Contact](#) [FAQs](#)

[f](#) [X](#) [in](#)

© 2024 RideHive. All rights reserved.

## Register as User:

RideHive

Home Login Register

### Register

Full Name

Email

Contact Number

Register As

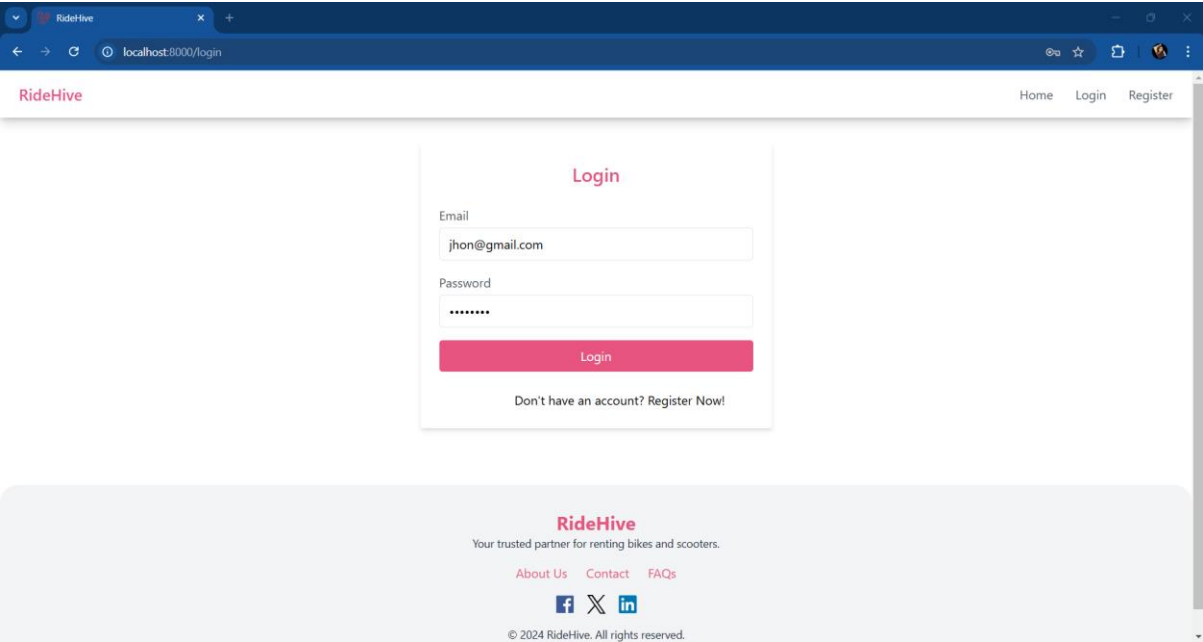
User

Password

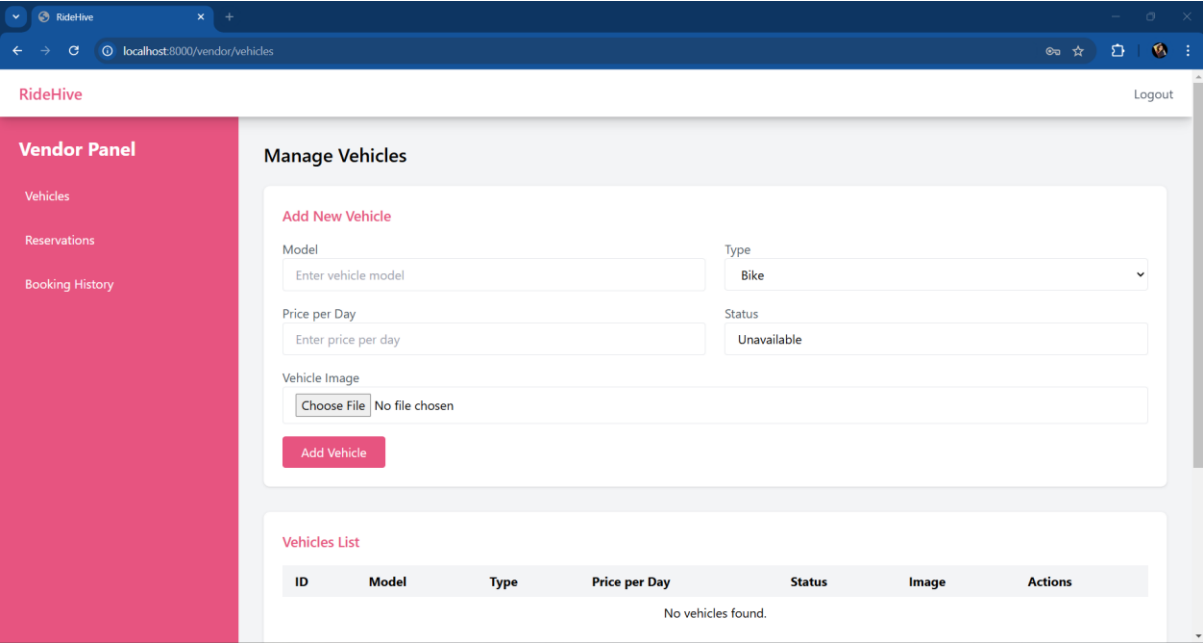
Confirm Password

Register

Login:



Vendor Panel:



## Adding Vehicle:

RideHive

Logout

Vehicles

Reservations

Booking History

Manage Vehicles

Add New Vehicle

Model

Scooter FX 10

Type

Scooter

Price per Day

30.5

Status

Unavailable

Vehicle Image

Choose File

scooter1.jpg

Add Vehicle

Vehicles List

ID	Model	Type	Price per Day	Status	Image	Actions
No vehicles found.						

RideHive

Logout

Vehicles

Reservations

Booking History

Vehicle added successfully.

Add New Vehicle

Model

Enter vehicle model

Type

Bike

Price per Day

Enter price per day

Status

Unavailable


Vehicle Image

Choose File

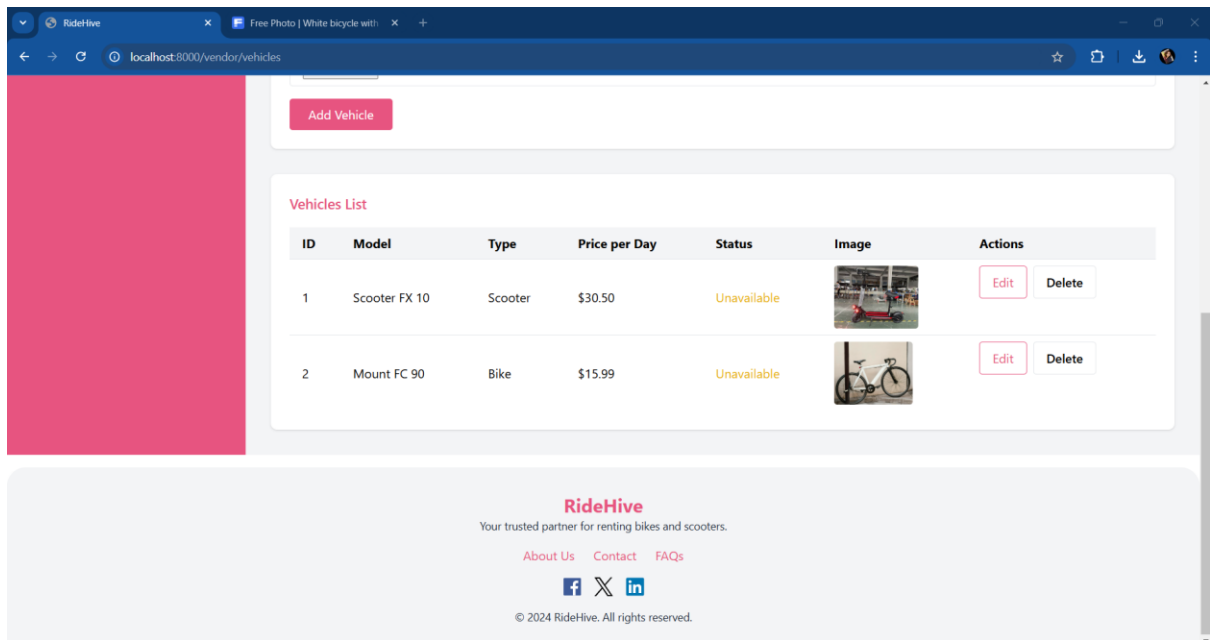
No file chosen

Add Vehicle

Vehicles List



ID	Model	Type	Price per Day	Status	Image	Actions
1	Scooter FX 10	Scooter	\$30.50	Unavailable		<div><div>Edit</div><div>Delete</div></div>





## Edit Vehicle:

Vehicles List

ID	Model	Type	Price per Day	Status	Image	Actions
1	Scooter FX 10	Scooter	\$30.50	Unavailable		<a href="#">Edit</a> <a href="#">Delete</a>
2	Mount FC 90	Bike	\$15.99	Unavailable		<a href="#">Edit</a> <a href="#">Delete</a>

Model

Mount FC 85

Type

Bike

Price per Day

15.99

Status

Unavailable

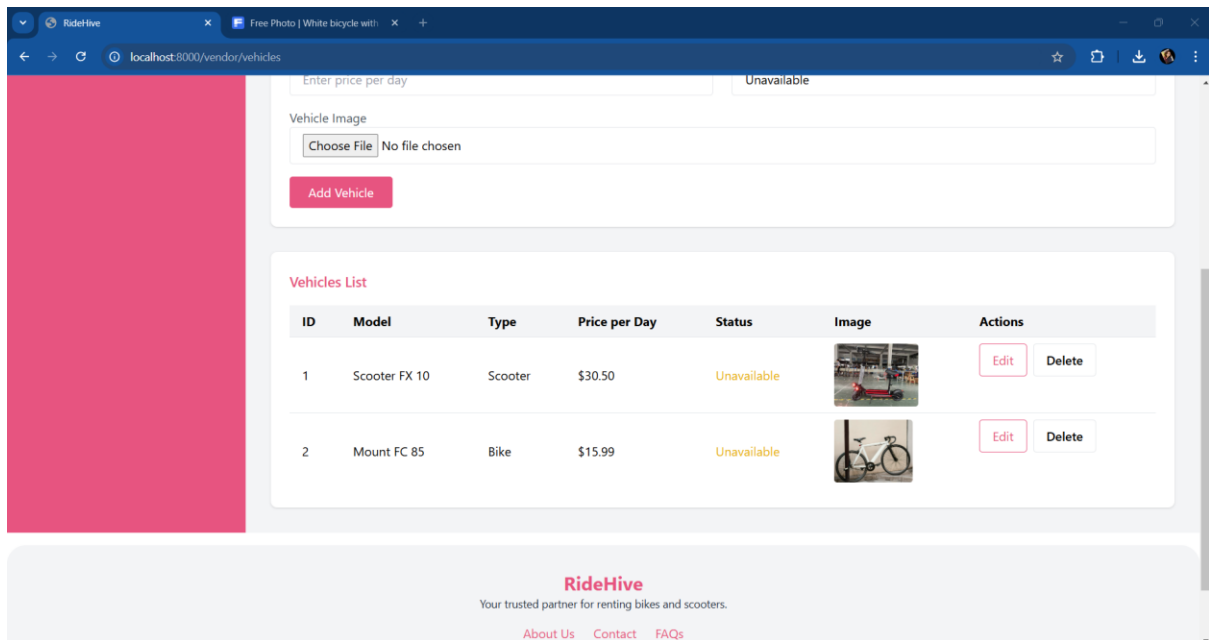
Vehicle Image

Choose File

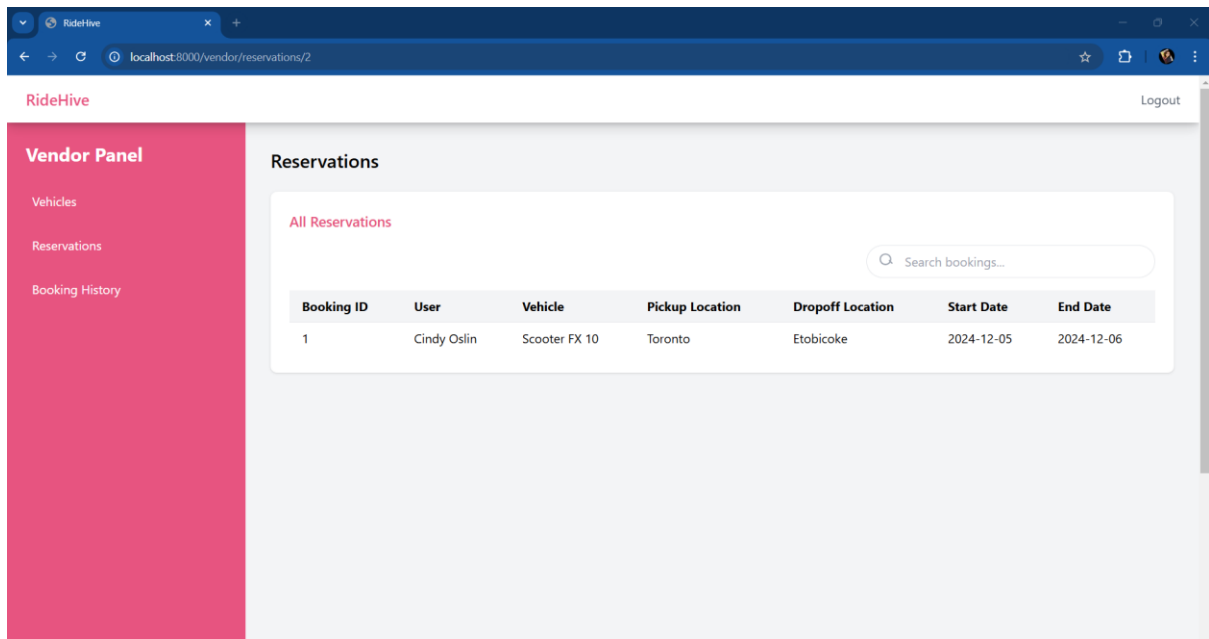
No file chosen

Save

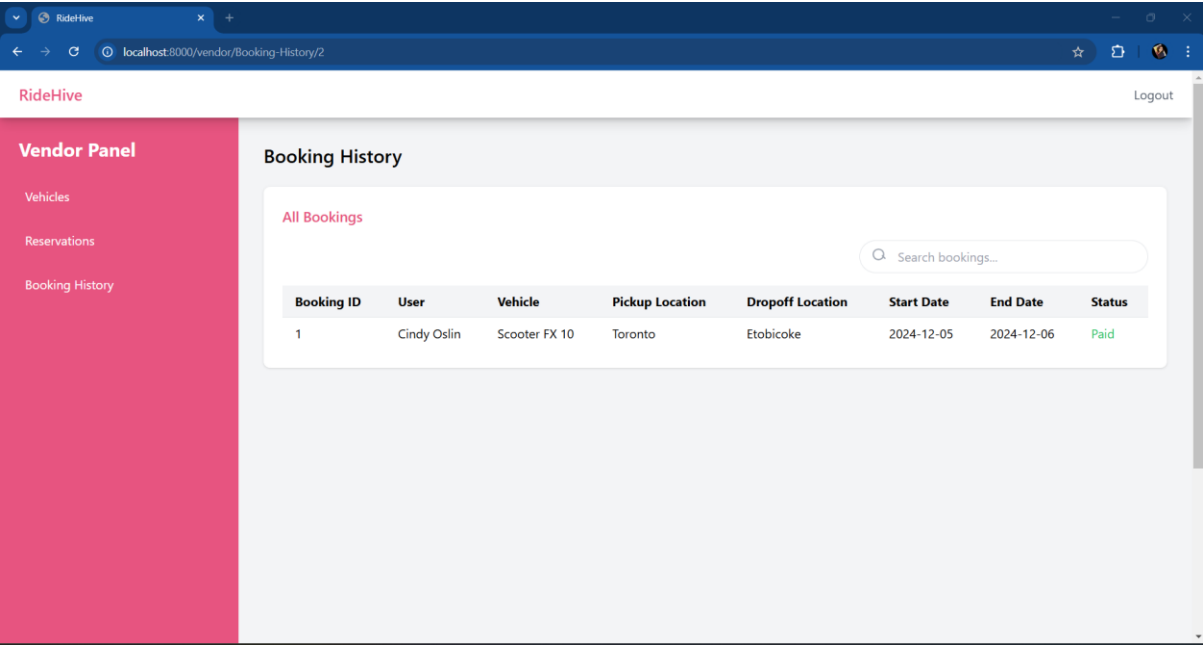
Cancel



## Reservations:

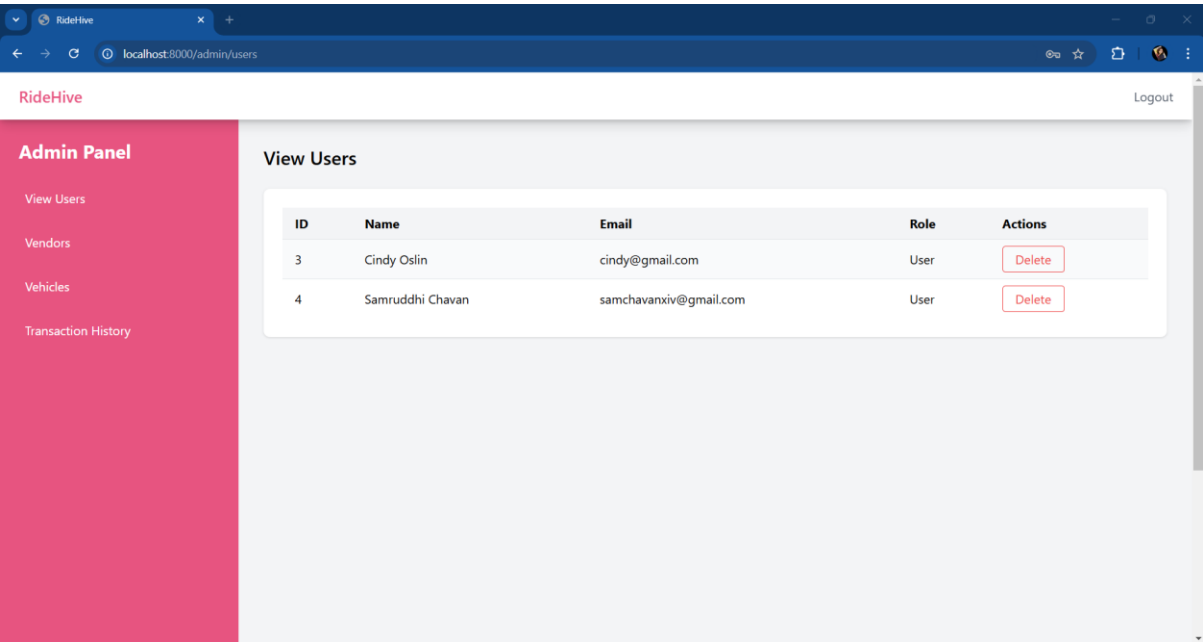


# Booking History:



# Admin Panel:

## User Management



# Vendor Management

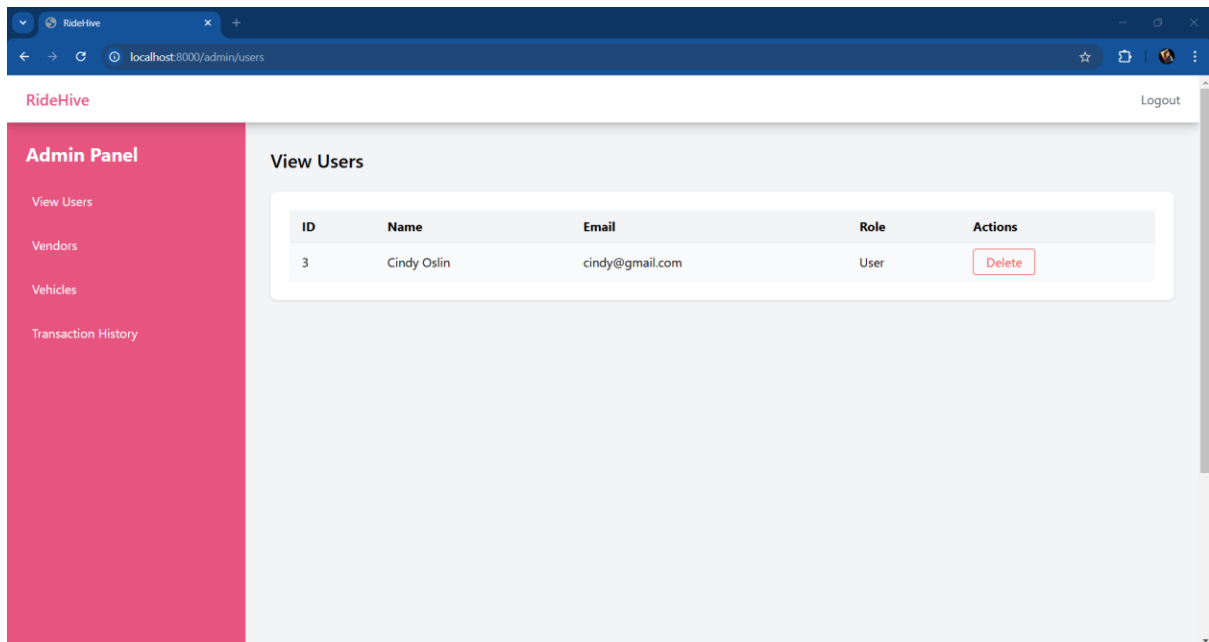
The screenshot shows the RideHive Admin Panel at the URL `localhost:8000/admin/vendors`. The left sidebar contains the 'Admin Panel' menu with options: View Users, Vendors, Vehicles, and Transaction History. The main content area is titled 'Manage Vendors' and displays a 'Vendors List' table.

ID	Name	Email	Contact	Actions
2	Jhon Markson	jhon@gmail.com	4537778595	<a href="#">Edit</a> <a href="#">Delete</a>

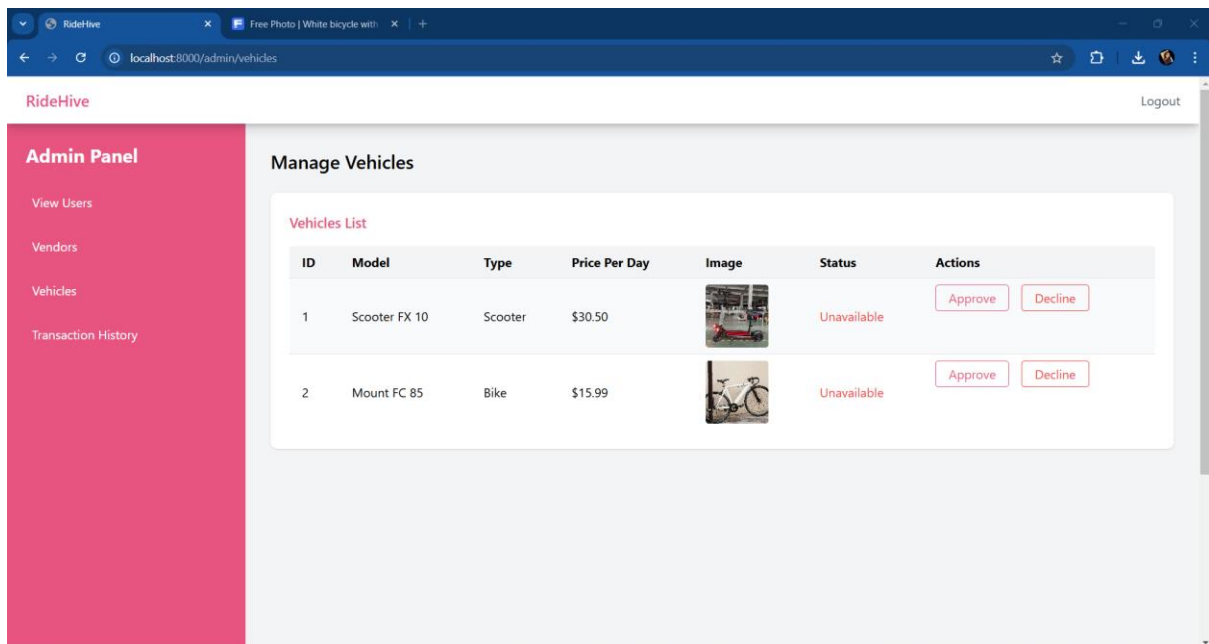
The screenshot shows the RideHive Admin Panel at the URL `localhost:8000/admin/users`. The left sidebar contains the 'Admin Panel' menu with options: View Users, Vendors, Vehicles, and Transaction History. The main content area is titled 'View Users' and displays a table of users. A confirmation dialog is overlaid on the table, asking for confirmation to delete a user.

ID	Name	Email	Role	Actions
3	Cindy Oslin	cindy@gmail.com	User	<a href="#">Delete</a>
4	Samruddhi Chavan	samchavanxiv@gmail.com	User	<a href="#">Delete</a>

localhost:8000 says  
Are you sure you want to delete this user?  
[OK](#) [Cancel](#)





## Admin Vehicle Approval Process:



## Transaction History:

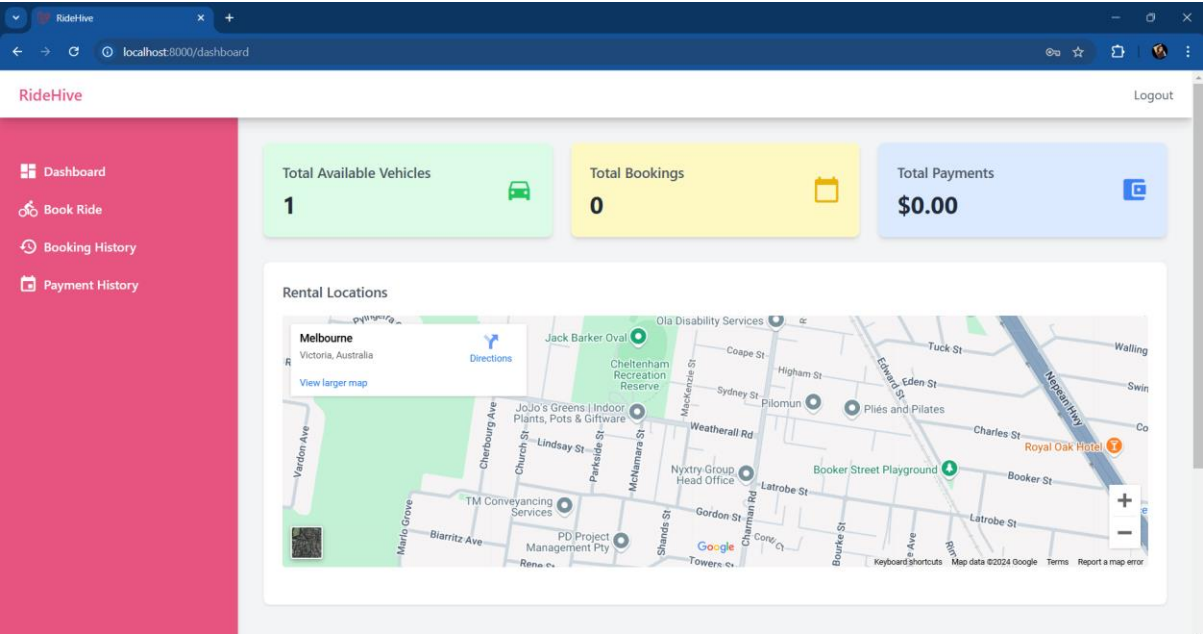
The screenshot shows the RideHive Admin Panel interface. On the left is a pink sidebar with the 'Admin Panel' header and a list of navigation items: 'View Users', 'Vendors', 'Vehicles', and 'Transaction History'. The main content area is titled 'Manage Vehicles' and contains a 'Vehicles List' table. The table has columns for ID, Model, Type, Price Per Day, Image, Status, and Actions. It lists two vehicles: a scooter and a bike. The scooter is 'Available' and the bike is 'Unavailable'. Each row has 'Approve' and 'Decline' buttons in the Actions column.

ID	Model	Type	Price Per Day	Image	Status	Actions
1	Scooter FX 10	Scooter	\$30.50		Available	<button>Approve</button> <button>Decline</button>
2	Mount FC 85	Bike	\$15.99		Unavailable	<button>Approve</button> <button>Decline</button>

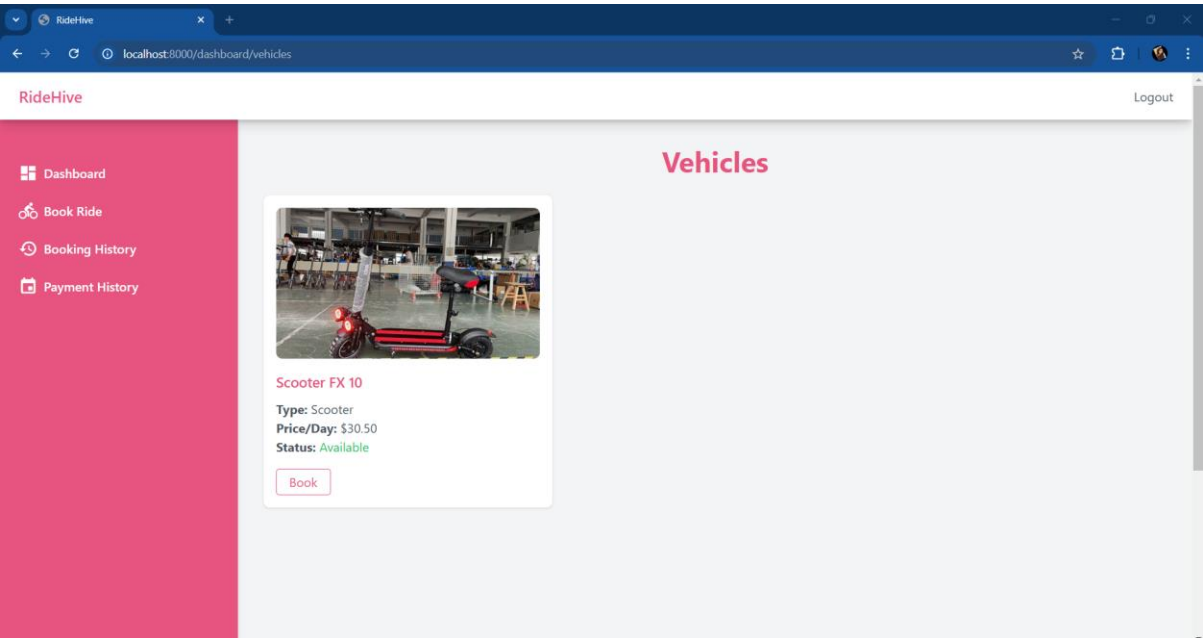
## Login as User:

The screenshot shows the RideHive User Login page. The header includes the 'RideHive' logo and navigation links for 'Home', 'Login', and 'Register'. The main content area features a 'Login' form with fields for 'Email' (containing 'cindy@gmail.com') and 'Password' (masked with dots). A pink 'Login' button is below the fields. A link 'Don't have an account? Register Now!' is at the bottom of the form. The footer contains the RideHive logo, the tagline 'Your trusted partner for renting bikes and scooters.', links for 'About Us', 'Contact', and 'FAQs', social media icons for Facebook, X, and LinkedIn, and the copyright notice '© 2024 RideHive. All rights reserved.'

# User Dashboard



## Available Vehicle Visible to user:



# Booking a Scooter:

RideHive

Logout

Ride Information

Pickup Location

Enter pickup location

Dropoff Location

Enter dropoff location

Start Date

dd-mm-yyyy

End Date

dd-mm-yyyy

Next

RideHive

Logout

Ride Information

Pickup Location

Toronto

Dropoff Location

Etobicoke

Start Date

05-12-2024

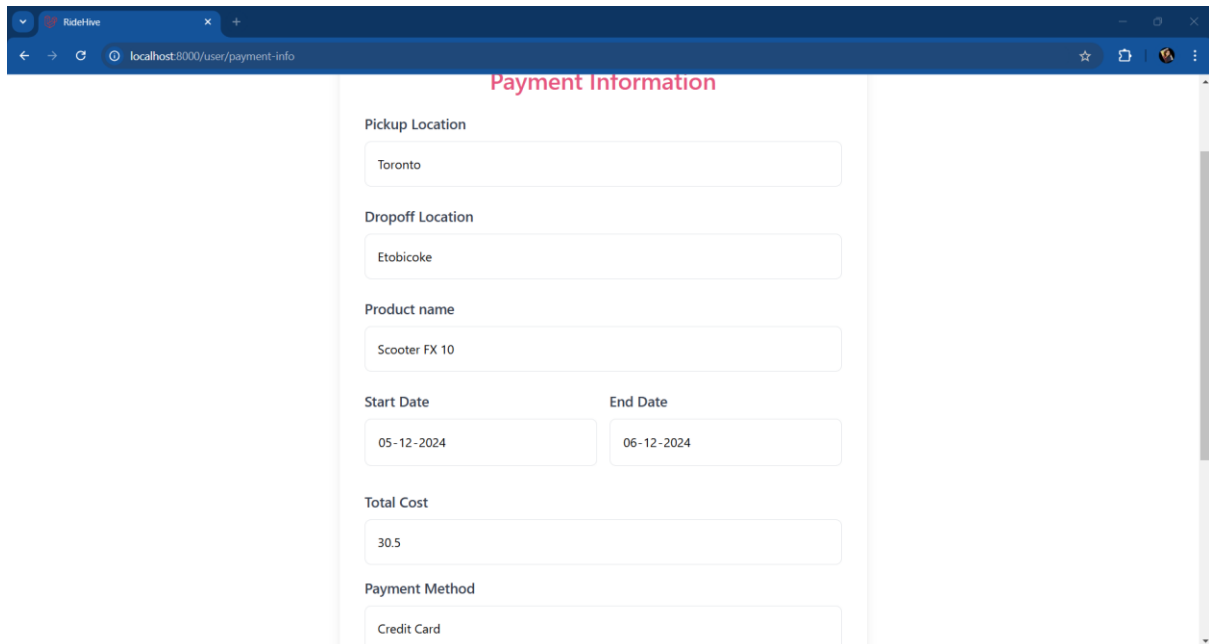
End Date

06-12-2024

Next



## Payment Information:



Payment Information

Pickup Location  
Toronto

Dropoff Location  
Etobicoke

Product name  
Scooter FX 10

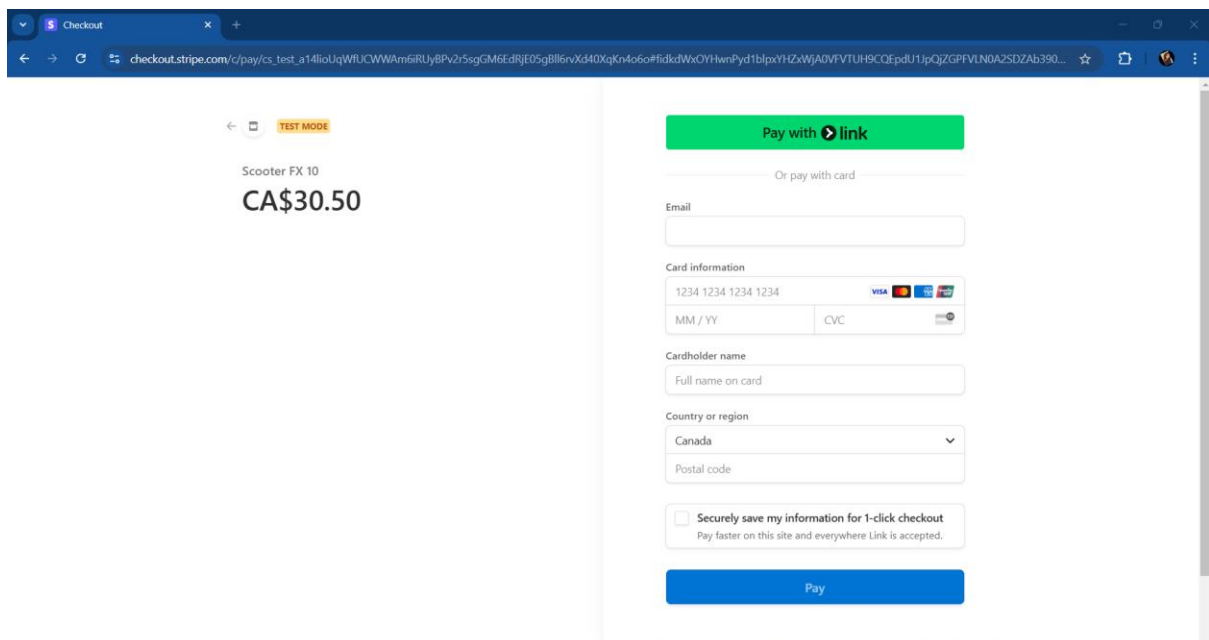
Start Date  
05-12-2024

End Date  
06-12-2024

Total Cost  
30.5

Payment Method  
Credit Card

## 3<sup>rd</sup> – Party API Integration with STRIPE:



TEST MODE

Scooter FX 10  
CA\$30.50

Pay with link

Or pay with card

Email

Card information  
1234 1234 1234 1234  
MM / YY CVC

Cardholder name  
Full name on card

Country or region  
Canada

Postal code

☐ Securely save my information for 1-click checkout  
Pay faster on this site and everywhere Link is accepted.

Pay


Checkout

checkout.stripe.com/c/pay/cs\_test\_a14loUqWfUCWWAm6iRUy8Pv2r5sgGM6EdRjE0SgBll6rvXd40XqKn4o6o#fidkdWxOYHwnPyd1blpxYHhZaWJhOjVFTU9BQCEpdU11pQjZGPVFN0A2SDZAb390...

TEST MODE

Scooter FX 10

CA\$30.50

Pay with  link

Or pay with card

Email

cindy@gmail.com

Card information

4242 4242 4242 4242

03 / 29299

VISA

Cardholder name

MS Samruddhi Chavan

Country or region

Canada

M4V 2S2

☐ Securely save my information for 1-click checkout

Pay faster on this site and everywhere Link is accepted.

Pay


Checkout

checkout.stripe.com/c/pay/cs\_test\_a14loUqWfUCWWAm6iRUy8Pv2r5sgGM6EdRjE0SgBll6rvXd40XqKn4o6o#fidkdWxOYHwnPyd1blpxYHhZaWJhOjVFTU9BQCEpdU11pQjZGPVFN0A2SDZAb390...

TEST MODE

Scooter FX 10

CA\$30.50

Pay with  link

Or pay with card

Email

cindy@gmail.com

Card information

4242 4242 4242 4242

03 / 29299

VISA

Cardholder name

MS Samruddhi Chavan


Country or region

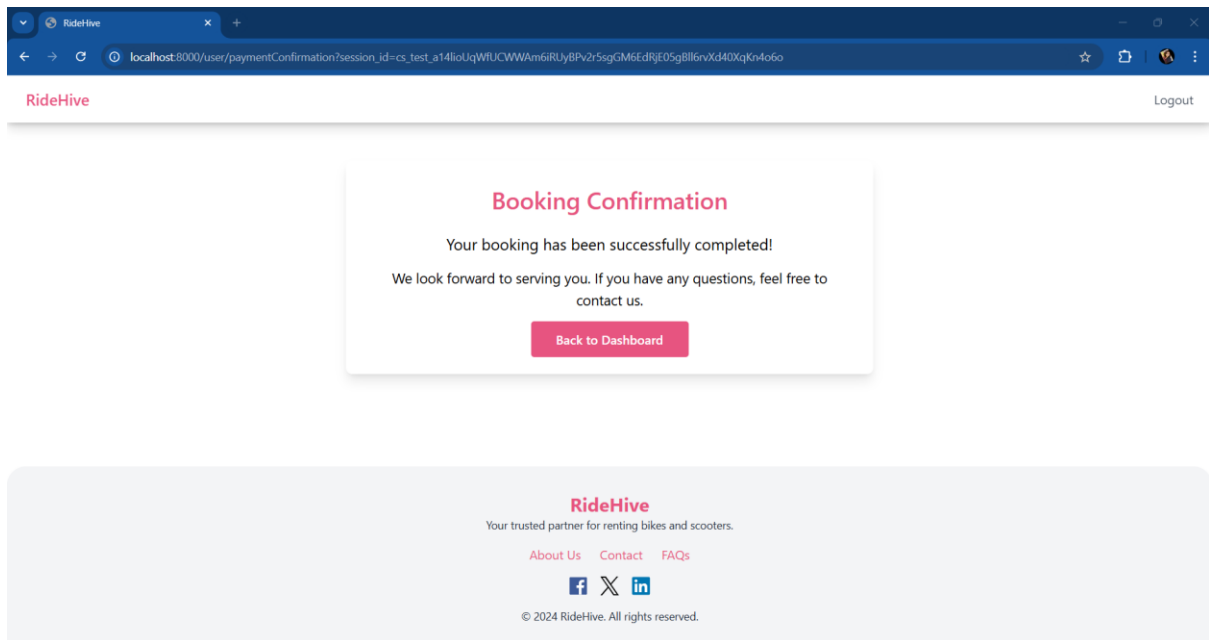
Canada

M4V 2S2

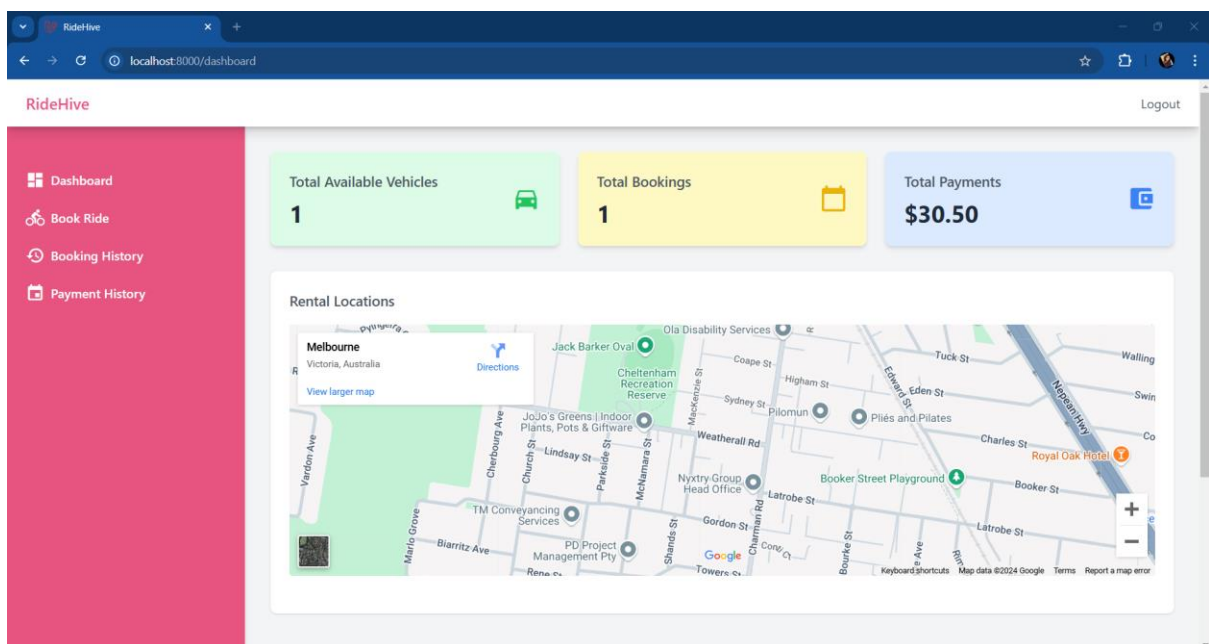
☐ Securely save my information for 1-click checkout

Pay faster on this site and everywhere Link is accepted.





## Updated Dashboard



# Booking History:

RideHive

Logout

Dashboard

Book Ride

Booking History

Payment History

HISTORY

Total Bookings

1

Completed

1

Pending

0

S. No	User	Vehicle	Pickup Location	Dropoff Location	Check-In	Check-Out	Status
1	Cindy Oslin	Scooter FX 10	Toronto	Etobicoke	05 Dec 2024, 12:00 AM	06 Dec 2024, 12:00 AM	Paid

# Payment History

RideHive

Logout

Dashboard

Book Ride

Booking History

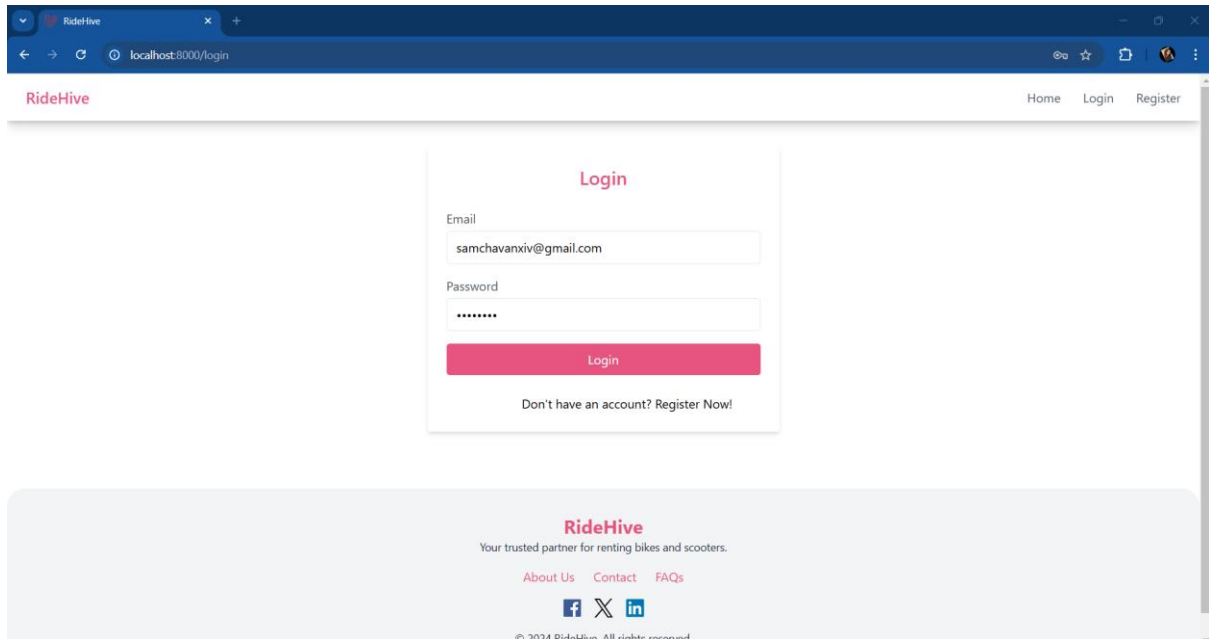
Payment History

Payment History

Payment ID	Booking ID	Amount	Payment Method	Payment Date	Status
#1	#1	\$30.50	Credit Card	2024-12-05	Successful

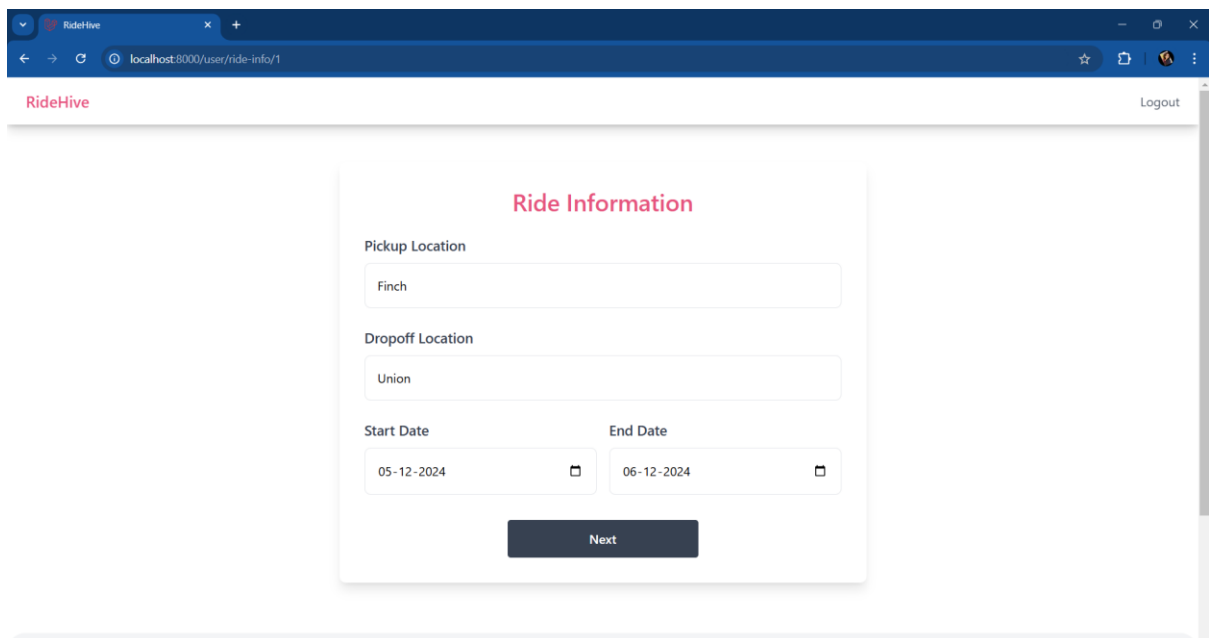
# Handling Overlap: Booking a Vehicle Already Reserved for the Requested Period

## Login as New User:



The screenshot shows the RideHive login page in a web browser. The browser's address bar displays 'localhost:8000/login'. The page features a 'Login' form with fields for 'Email' (containing 'samchavanxiv@gmail.com') and 'Password' (masked with dots). A pink 'Login' button is positioned below the password field. A link 'Don't have an account? Register Now!' is located at the bottom of the form. The footer includes the RideHive logo, the tagline 'Your trusted partner for renting bikes and scooters.', links for 'About Us', 'Contact', and 'FAQs', social media icons for Facebook, X, and LinkedIn, and a copyright notice '© 2024 RideHive. All rights reserved.'

## Book the Scooter:



The screenshot shows the 'Ride Information' booking page in a web browser. The browser's address bar displays 'localhost:8000/user/ride-info/1'. The page features a 'Ride Information' form with fields for 'Pickup Location' (containing 'Finch'), 'Dropoff Location' (containing 'Union'), 'Start Date' (containing '05-12-2024'), and 'End Date' (containing '06-12-2024'). Each date field includes a calendar icon. A dark blue 'Next' button is located at the bottom of the form. The top right corner of the page has a 'Logout' link.

### Booking Confirmation

Booking Error!

Vehicle already booked. Please choose another vehicle.

Back to Dashboard

## 10. Conclusion

The RideHive Bike and Scooter Rental System is a well-designed platform addressing the growing need for affordable and flexible urban transportation. It provides a seamless experience for users to register, log in, search, and book vehicles, while vendors can list and manage their fleets efficiently. The integration of Stripe ensures secure online payment processing, while Laravel's MVC architecture enhances modularity and scalability. The database is robustly structured, linking users, vehicles, bookings, payments to maintain operational efficiency. Additionally, an admin dashboard facilitates comprehensive monitoring and management of users, vendors, transactions, and bookings. By leveraging modern technologies like Laravel, MySQL, and Tailwind CSS, RideHive is equipped to handle current demands and future enhancements, making it a scalable and secure rental solution.